A Short Tutorial on Model-Based Diagnosis*

Claudia Picardi

Diagnosis: from Greek *diagnOsis*, from *diagignOskein* to distinguish, from *dia-* + *gignOskein* to know. *The art or act of identifying a disease from its signs and symptoms.*
[from Merriam Webster Collegiate Dictionary - http://www.m-w.com]

1 Introduction and early history

Model-based diagnosis — often referred to as MBD for short — denotes an area of Artificial Intelligence which:

- Tackles the problem of diagnosing a system (a designed artifact), that is, tracking the causes (faults) responsible for a system failure. Usually the input of a diagnosis algorithm is a set of symptoms, i.e. discrepancies between the observed behaviour and the expected one, and its output is a set of possible faults that can have caused such symptoms.
- Performs diagnosis starting from a model of the system, describing how the system is supposed to behave (correct behaviour), or the relations between faults and symptoms (faulty behaviour), possibly both.

The model-based approach to diagnosis started to be investigated by A.I. researchers in the late seventies, as a possible alternative to the expert-system approach [9]. At that time diagnosis was one of the largest categories of expert systems in use; nonetheless the use of expert knowledge in diagnosis started to show some major limitation [18], especially when applied to diagnosis of artifacts rather than medical diagnosis. The acquiring and maintenance of the needed expert knowledge became a bottleneck in the deployment of diagnostic expert systems.
Thus researchers started investigating ([11], [3], [25], [10], [15]) how to exploit, rather than expert knowledge, the then called *deep knowledge*. The idea was to exploit objective information about the system behaviour (whereas expert knowledge could be regarded as the expert's subjective view of the system), that could be available for other purposes than diagnosis. In the case of engineered artifacts, this corresponded to exploiting models of the diagnosed system that had been created while designing it.
Design models usually adopt a component-oriented approach, that is, they describe a system in terms of its components and the way these components interact. Moreover, since such models are not specifically designed for diagnosis, they do not include knowledge about how the system behaves in the presence of faults. Therefore many of the early works cited above started from the assumption that *(i)* the model contained information about the sole correct behaviour of the system, and *(ii)* that the task of diagnosis was to identify which *components* of the system were broken or mis-functioning. This approach, systematized by Reiter [24], is the one known as *consistency-based diagnosis*.

*This is an evolving document, and by no means a complete account of the literature on model-based diagnosis. If you have any comments and/or corrections please contact me at picardi@di.unito.it

General-purpose, component-oriented models are not however the only possible "deep knowledge" about a system. Many other researchers ([19], [23], [17], [13], [20], [27], [2], [4], [6], [21]), while agreeing to the need of overcoming the limitations in the use of expert knowledge, adopted a different approach which involved describing objective knowledge about the causal relationships between faults and symptoms [1]. They thus exploited the *causal model* of a system, containing explicit information about which faults can occur and which chain of consequences they provoke, up to their observable manifestations. This line of research led to the definition of *abductive diagnosis* [8].
The two approaches eventually converged into the parallel ideas of exploiting information about faults in consistency based diagnosis, and information about correct behaviour in abductive diagnosis. Including in a component-oriented model information about faults corresponds to describing, along with the correct behaviour of system components, also their possible faults and their consequences. Thus, models of correct behaviour started to be endowed with *fault models*, and causal models started to include a description of nominal behaviour ([22], [26]). Moreover, the two approaches — consistency-based and abductive — were integrated ([14], [5], [12]) and shown to be the extremes of a wide spectrum of possible definitions of diagnosis ([7]).
At the beginning of the 90s the early history of MBD ends: at that point the main results were consolidated (an in-depth account of those works can be found in [16]). MBD - at least at its basic level - started to move towards industrial applications, which would turn out to be the source of new challenges and of a renewed interest in many aspects that until then had been overlooked in favour of foundational issues.

2 Consistency-based diagnosis

2.1 Definition and basic properties

In this section we will introduce the notion of *diagnostic problem* and characterize formally the notion of *consistency-based diagnosis*. Mostly we will follow the characterization in first-order logic proposed in the seminal paper *"A Theory of Diagnosis from First Principles"* written by Raymond Reiter and published on the *Artificial Intelligence* journal in 1987 [24].
The idiom "First Principles" denotes an objective knowledge on the system that has to be diagnosed, possibly available independently from the diagnostic task. Such knowledge is what we call the *model* of the system, and, as we assume the model has been not designed explicitly for the diagnostic task, it expresses how the system behaves when it is working properly.
The goal of diagnosis is to find, given some unexpected behaviour, the part (or *component*) of the system responsible for it. Thus we expect the system model to be decompsable in components, in a way that allows to ascribe to each component the responsibility for a specific part of the behaviour. This comes quite natural when we thing of engineered artifacts, that are usually designed and built by connecting components. Therefore, we can say that a system description (or system model) is made of three parts:

- **Behaviour of component types**. For each type of component, we describe the expected correlation between its inputs and outputs, under the assumption that the component is working correctly. The description of a component type has a well defined structure. In particular all formulas describing the behaviour follow the template:

$$type_i(x) \wedge ok(x) \rightarrow \Phi(x)$$

  where $\Phi(x)$ is a generic formula, $type_i$ is a predicate stating that component x belongs to type i, and the predicate $ok(x)$ states that the component x is behaving normally. Thus the

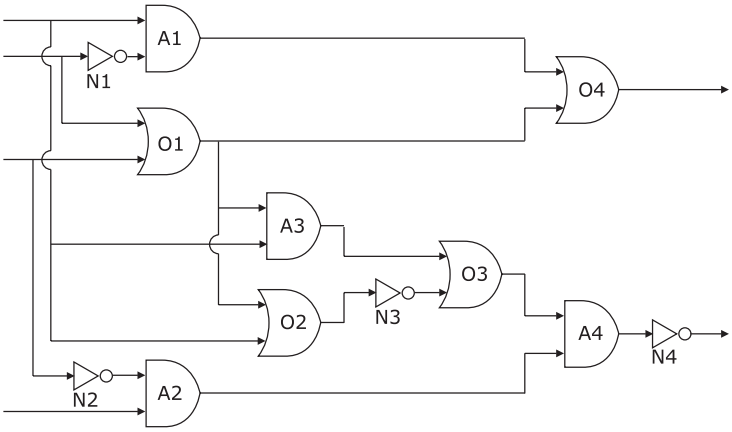

Figure 1: Structure of circuit CIRC_1.

overall meaning is "if a component has type i and it is behaving normally, then it behaves as described by $\Phi(x)$". Implicitly, we are also saying that when the component is *not* behaving normally, it could do anything.

- **List of components**. This is a set of statements that list the components in the systems and state their type. Each statement is a ground formula of the kind:

$$type_i(c)$$

where c is a constant representing an individual component (we can see c as the component name). The meaning is "component c is of type i (and thus, if it behaves correctly, we can apply to it the formulas describing the behaviour for type i)".

- **System structure**. A set of statements describing the connections between inputs and outputs of different components. Typically these are equality statements saying that a given input of a component is equal to a given output of another component:

$$in_i(c) = out_j(d)$$

where c and d are components' names and in_i and out_j are function names denoting respectively one of c's inputs and one of d's outputs.

Example 1 Let us consider the circuit depicted in figure 1. We can distinguish three component types (*and*, *or* and *not* gates), and twelve components (four for each type). First we describe the behaviour of each component type:

$$\begin{aligned}
\mathsf{TYPES}_1 = \quad & \{(not_g(x) \wedge ok(x)) \rightarrow \quad (in(x) = 0 \vee in(x) = 1) \wedge \\
& \qquad\qquad\qquad\qquad\quad (in(x) = 1 \rightarrow out(x) = 0) \wedge \\
& \qquad\qquad\qquad\qquad\quad (in(x) = 1 \rightarrow out(x) = 0), \\
& (and_g(x) \wedge ok(x)) \rightarrow \quad (in_1(x) = 0 \vee in_1(x) = 1) \wedge \\
& \qquad\qquad\qquad\qquad\quad (in_2(x) = 0 \vee in_2(x) = 1) \wedge \\
& \qquad\qquad\qquad\qquad\quad ((in_1(x) = 0 \vee in_2(x) = 0) \rightarrow out(x) = 0) \wedge \\
& \qquad\qquad\qquad\qquad\quad ((in_1(x) = 1 \wedge in_2(x) = 1) \rightarrow out(x) = 1), \\
& (or_g(x) \wedge ok(x)) \rightarrow \quad (in_1(x) = 0 \vee in_1(x) = 1) \wedge \\
& \qquad\qquad\qquad\qquad\quad (in_2(x) = 0 \vee in_2(x) = 1) \wedge \\
& \qquad\qquad\qquad\qquad\quad ((in_1(x) = 0 \wedge in_2(x) = 0) \rightarrow out(x) = 0) \wedge \\
& \qquad\qquad\qquad\qquad\quad ((in_1(x) = 1 \vee in_2(x) = 1) \rightarrow out(x) = 1)\}
\end{aligned}$$

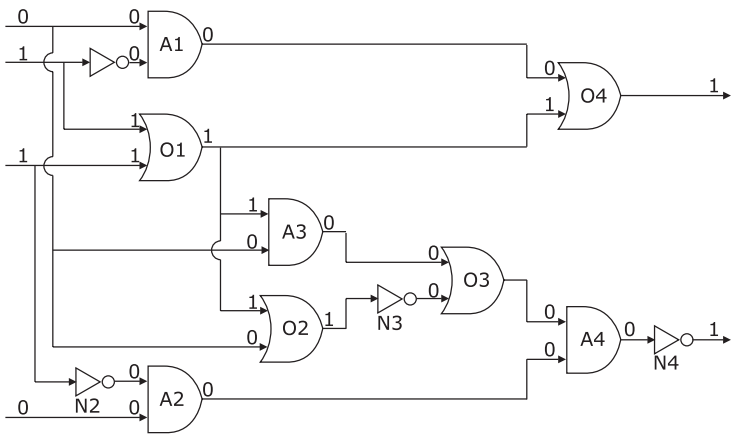

Figure 2: Logical consequences of $\mathsf{SD}_1 \cup \mathsf{COMPS} \cup \mathsf{OBS}_{in}$.

Then we list the components of our system, stating their types:

$$\mathsf{COMPTYPES}_1 = \{not\_g(\mathsf{N1}), not\_g(\mathsf{N2}), not\_g(\mathsf{N3}), not\_g(\mathsf{N4}), and\_g(\mathsf{A1}), and\_g(\mathsf{A2}), and\_g(\mathsf{A3}), and\_g(\mathsf{A4}), or\_g(\mathsf{O1}), or\_g(\mathsf{O2}), or\_g(\mathsf{O3}), or\_g(\mathsf{O4})\}$$

Finally, we state the connections as depicted in figure 1.

$$\begin{array}{lll} \mathsf{CONN}_1 = \{ \quad in(\mathsf{N1}) = in_1(\mathsf{O1}), & in_2(\mathsf{O1}) = in(\mathsf{N2}), & out(\mathsf{N1}) = in_2(\mathsf{A1}), \\ out(\mathsf{N2}) = in_1(\mathsf{A2}), & in_1(\mathsf{A1}) = in_2(\mathsf{A3}), & in_1(\mathsf{A1}) = in_2(\mathsf{O2}), \\ out(\mathsf{O1}) = in_1(\mathsf{A3}), & out(\mathsf{O1}) = in_1(\mathsf{O2}), & out(\mathsf{O1}) = in_2(\mathsf{O4}), \\ out(\mathsf{A1}) = in_1(\mathsf{O4}), & out(\mathsf{O2}) = in(\mathsf{N3}), & out(\mathsf{A3}) = in_1(\mathsf{O3}), \\ out(\mathsf{N3}) = in_2(\mathsf{O3}), & out(\mathsf{O3}) = in_1(\mathsf{A4}), & out(\mathsf{A2}) = in_2(\mathsf{A4}), \\ out(\mathsf{A4}) = in(\mathsf{N4})\} & & \end{array}$$

Thus, the complets model of circuit CIRC_1 is:

$$\mathsf{SD}_1 = \mathsf{TYPES}_1 \cup \mathsf{COMPTYPES}_1 \cup \mathsf{CONN}_1 \qquad \bigstar$$

Diagnosis is needed when there is a discrepancy between the *observed* behaviour of a system and its *expected* behaviour, as described in the model. Given the notion of model we provided, we can say that we have a **diagnostic problem** when our observations on the system are *inconsistent* with the assumption that all the system components are working correctly. Let us formalize this notion.

Definition 2 A **diagnostic problem** is a triple

$$\langle \mathsf{SD}, \mathsf{COMPS}, \mathsf{OBS} \rangle$$

where

- SD is a system description;
- COMPS is a set of component names mentioning the components that can be faulty;
- OBS is a set of ground atomic formulas expressing the observations made on the system, such that the set of formulas $\mathsf{SD} \cup \{ok(c) \mid c \in \mathsf{COMPS}\} \cup \mathsf{OBS}$ is *inconsistent*. $\bigstar$

Example 3 Let us consider again our circuit example, and let us consider the following diagnostic problem:

$$\mathsf{DP} = \langle \mathsf{SD}_1, \mathsf{COMPS}, \mathsf{OBS} \rangle$$

where

- SD_1 is as described previously;
- COMPS contains all components depicted in figure 1;
- OBS contains observations regarding the system global inputs and outputs:

$$\begin{aligned} \mathsf{OBS} = \{ \quad & in_1(\mathsf{A1}) = 0, \quad in(\mathsf{N1}) = 1, \quad in_2(\mathsf{O1}) = 1, \\ & in_2(\mathsf{A2}) = 0, \quad out(\mathsf{O4}) = 1, \quad out(\mathsf{N4}) = 0\} \end{aligned}$$

Given the assumption that all the components are *ok*, and given the observations on input values, we are able to derive what are the expected system outputs. Here is an example of a simple derivation:

$in(\mathsf{N1}) = 1, in(\mathsf{N1}) = in_1(\mathsf{O1})$	$\vdash$	$in_1(\mathsf{O1}) = 1$
$or\_g(O1), ok(O1),$ $(or\_g(x) \wedge ok(x)) \rightarrow ((in_1(x) = 1 \vee in_2(x) = 1) \rightarrow out(x) = 1)$	$\vdash$	$(in_1(O1) = 1 \vee in_2(O1) = 1) \rightarrow$ $\rightarrow out(O1) = 1$
$in_1(O1) = 1, (in_1(O1) = 1 \vee in_2(O1) = 1) \rightarrow out(O1) = 1$	$\vdash$	$out(O1) = 1$
$out(O1) = 1, out(\mathsf{O1}) = in_2(\mathsf{O4})$	$\vdash$	$in_2(\mathsf{O4} = 1$
$or\_g(O4), ok(O4),$ $(or\_g(x) \wedge ok(x)) \rightarrow ((in_1(x) = 1 \vee in_2(x) = 1) \rightarrow out(x) = 1)$	$\vdash$	$(in_1(O4) = 1 \vee in_2(O4) = 1) \rightarrow$ $\rightarrow out(O4) = 1$
$in_2(\mathsf{O4} = 1, (in_1(O4) = 1 \vee in_2(O4) = 1) \rightarrow out(O4) = 1$	$\vdash$	$out(O4) = 1$

Figure 2 shows all the values that can be derived as logical consequences from $\mathsf{SD}_1 \cup \mathsf{COMPS} \cup \mathsf{OBS}_{in}$, where OBS_{in} denotes the set of observations restricted to input values. In particular, we see that $\mathsf{SD}_1 \cup \mathsf{COMPS} \cup \mathsf{OBS}_{in} \vdash out(\mathsf{N4}) = 1$. Since OBS contains the statement $out(\mathsf{N4}) = 0$ it is straightforward to see that $\mathsf{SD}_1 \cup \mathsf{COMPS} \cup \mathsf{OBS}$ is inconsistent. ★

Now that we defined the notion of diagnostic problem, we have to define its solution; in other words we need to say what a *diagnosis* is. In general, a diagnosis Δ is a set of components that we assume to be faulty. Thus Δ is a subset of COMPS: 2^{COMPS} is the search space for a diagnostic algorithm.
But when does a candidate diagnosis become a correct solution for a given diagnostic problem? Intuitively, the inconsistency that characterizes a diagnostic problem is caused by the assumption that all components are *ok*, which is obviously false. Now suppose that by removing the assumption $ok(c)$ for a given component c, and adding instead the assumption $\neg ok(c)$, we have that the set of formulas is brought back to consistency: then we can argue that assuming that c is broken *explains* the inconsistency, thus $\Delta = \{c\}$ is a diagnosis for the diagnostic problem.
We can formalize this as follows:

Definition 4 Let $\mathsf{DP} = \langle \mathsf{SD}, \mathsf{COMPS}, \mathsf{OBS} \rangle$ be a diagnostic problem. We say that a set $\Delta \subseteq \mathsf{COMPS}$ is a **consistency-based diagnosis** for DP if it is a *minimal* set such that $\mathsf{SD} \cup \{ok(c) \mid c \in \mathsf{COMPS} \setminus \Delta\} \cup \{\neg ok(c) \mid c \in \Delta\} \cup \mathsf{OBS}$ is consistent. ★

The reason why we call this a *consistency-based* diagnosis is that with this definition we are saying that explaining an inconsistent observation corresponds to restoring consistency.
Before giving an example of diagnosis, let us prove some useful properties.

Proposition 5 *Let DP denote a diagnostic problem, and $\Delta = \{c_1, \ldots, c_k\}$ denote a consistency-based diagnosis for DP. Then*

$$\mathit{SD} \cup \mathit{OBS} \cup \{ok(c) \mid c \in \mathit{COMPS} \setminus \Delta\} \models \neg ok(c_i)$$

for all $c_i \in \Delta$.

Proof. Let us assume that Δ is a diagnosis for DP but that the above claim is not true. Then:

$$\mathsf{SD} \cup \mathsf{OBS} \cup \{ok(c) \mid c \in \mathsf{COMPS} \setminus \Delta\} \nvDash \neg ok(c_1) \wedge \ldots \wedge \neg ok(c_k) \equiv \neg(ok(c_1) \vee \ldots \vee ok(c_k))$$

This is equivalent to saying that

$$\mathsf{SD} \cup \mathsf{OBS} \cup \{ok(c) \mid c \in \mathsf{COMPS} \setminus \Delta\} \cup \{ok(c_1) \vee \ldots \vee ok(c_k)\}$$

is consistent.
This means that there is a combination of correctness/incorrectness assumptions regarding the components in Δ (with at least one component assumed correct, and at least one assumed incorrect) that is consistent with $\mathsf{SD} \cup \mathsf{OBS} \cup \{ok(c) \mid c \in \mathsf{COMPS} \setminus \Delta\}$. More formally said, it must be possible to partition Δ into two non empty sets $\{c_{i_1}, \ldots, c_{i_m}\}$ and $\{c_{j_1}, \ldots, c_{j_{k-m}}\}$ such that

$$\mathsf{SD} \cup \mathsf{OBS} \cup \{ok(c) \mid c \in \mathsf{COMPS} \setminus \Delta\} \cup \{ok(c_{i_1}) \wedge \ldots \wedge ok(c_{i_m})\} \cup \{\neg ok(c_{j_1}) \wedge \ldots \wedge \neg ok(c_{j_{n-m}})\}$$

is consistent.
Then $\Delta' = \{c_{j_1}, \ldots, c_{j_{k-m}}\}$ is a subset of Δ such that

$$\mathsf{SD} \cup \mathsf{OBS} \cup \{ok(c) \mid c \in \mathsf{COMPS} \setminus \Delta'\} \cup \{\neg ok(c) \mid c \in \Delta'\}$$

is consistent. But this contradicts the hypothesis that Δ is a diagnosis for DP, because it violates the minimality requirement. Therefore the thesis must hold. ★.
A consequence of this proposition is a simplification of the definition of diagnosis:

Proposition 6 $\Delta \subseteq$ *COMPS is a consistency-based diagnosis for a diagnostic problem DP if and only if it is a minimal set such that* $\mathsf{SD} \cup \{ok(c) \mid c \in \mathsf{COMPS} \setminus \Delta\} \cup \mathsf{OBS}$ *is consistent.*

Proof. This proposition essentially states that *incorrectness* assumptions are redundant in the definition of diagnosis. We divide the proof in two.

i. *If* Δ *is a diagnosis then it is also a minimal set such that* $\mathsf{SD} \cup \{ok(c) \mid c \in \mathsf{COMPS} \setminus \Delta\} \cup \mathsf{OBS}$ *is consistent.*

   If Δ is a diagnosis then $\mathsf{SD} \cup \{ok(c) \mid c \in \mathsf{COMPS} \setminus \Delta\} \cup \{\neg ok(c) \mid c \in \Delta\} \cup \mathsf{OBS}$ is consistent. Removing formulas from a consistent set preserves consistency, thus also $\mathsf{SD} \cup \{ok(c) \mid c \in \mathsf{COMPS} \setminus \Delta\} \cup \mathsf{OBS}$ is consistent. Moreover, there is no larger consistent set: in fact, by proposition 5 we have that $\mathsf{SD} \cup \{ok(c) \mid c \in \mathsf{COMPS} \setminus \Delta\} \cup \mathsf{OBS} \vDash \neg ok(c_i)$ for all $c_i \in \Delta$, thus adding further correctness assumptions would necessarily lead to inconsistency.

ii. *If* Δ *is a minimal set such that* $\mathsf{SD} \cup \{ok(c) \mid c \in \mathsf{COMPS} \setminus \Delta\} \cup \mathsf{OBS}$ *is consistent, then* Δ *is a diagnosis.*

   The minimality of Δ implies that $\mathsf{SD} \cup \{ok(c) \mid c \in \mathsf{COMPS} \setminus \Delta\} \cup \mathsf{OBS} \cup \{ok(c_i)\}$ is inconsistent for any $c_i \in \Delta$. Thus for all $c_i \in \Delta$ we have that $\mathsf{SD} \cup \{ok(c) \mid c \in \mathsf{COMPS} \setminus \Delta\} \cup \mathsf{OBS} \vDash \neg ok(c_i)$ and therefore $\mathsf{SD} \cup \{ok(c) \mid c \in \mathsf{COMPS} \setminus \Delta\} \cup \{\neg ok(c) \mid c \in \Delta\} \cup \mathsf{OBS}$ is consistent. In order to prove that Δ is a diagnosis we still have to show that it is minimal. But if there existed a diagnosis $\Delta' \subset \Delta$, for the first part of this proof we would get that $\mathsf{SD} \cup \{ok(c) \mid c \in \mathsf{COMPS} \setminus \Delta'\} \cup \mathsf{OBS}$ is consistent, thereby contradicting the hypothesis of this proposition. ★

Finally, we have the following corollary:

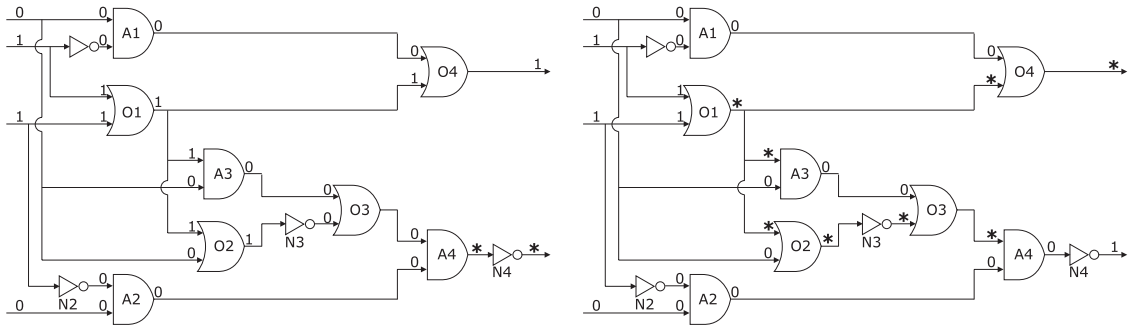

Figure 3: (a) Assuming that A4 is faulty. (b) Assuming that O1 is faulty.

Proposition 7 *Given a diagnostic problem DP, a diagnosis exists for it if and only if* $SD \cup OBS$ *is consistent.*

Proof. This proposition follows immediately from the simplified definition of diagnosis. In fact, if $\mathsf{SD} \cup \mathsf{OBS}$ is inconsistent, every other set containing it is inconsistent as well. In particular, for every possible $\Delta \subseteq \mathsf{COMPS}$, we have that $\mathsf{SD} \cup \{ok(c) \mid c \in \mathsf{COMPS} \setminus \Delta\} \cup \mathsf{OBS}$ is inconsistent; thus no diagnosis can exist. On the other hand, if we take $\Delta \equiv \mathsf{COMPS}$, we have $\mathsf{SD} \cup \{ok(c) \mid c \in \mathsf{COMPS} \setminus \Delta\} \cup \mathsf{OBS} \equiv \mathsf{SD} \cup \mathsf{COMPS}$, thus if this latter set is consistent then either COMPS itself is a diagnosis, or one of its subsets is. ★

Notice that the minimality requirement in the definition of diagnosis is strictly necessary: in fact, if Δ restores consistency, every Δ' containing Δ does as well. If we remove the minimality requirement, we get many redundant diagnoses, while the implicit idea is that if an observation can be explained by assuming that c alone is broken, there is no point in assuming that another component c' is broken as well.

Example 8 Let us find diagnoses for the diagnostic problem defined in the previous example. We can notice that removing an *ok* assumption about a component actually means *removing* that component's behaviour from the system description, which in turn means assuming that its output is unknown and can have any value.

For example, let us suppose we remove the assumption $ok(\mathsf{A4})$. Figure 3.a shows what happens: some values in the circuit, starting from the output of A4 are unknown (represented in figure by a *). Since the output of A4, that goes in input to N4, is unknown, also the output of N4 is. In other words, the model "without" A4 does not imply anymore that $out(\mathsf{N4}) = 1$. Thus consistency with observation $out(\mathsf{N4}) = 0$ is restored and $\{\mathsf{A4}\}$ is a diagnosis for our diagnostic problem.

On the other hand, let us remove the assumption $ok(\mathsf{O1})$ (results are depicted in figure 3.b). Since the output of O1 is unknown, we have that O2 receives in input a 0 and an unknown value. The output of O2 is thus unknown as well. On the other hand, A3 receives in input a 0 and an unknown value, but for an and gate this is sufficient to decide that the output is 0. Then, also O3 receives in input a 0 and an unknown value, so also the output of O3 is unknown. Finally, A4 receives in input a 0 and an unknown value, but exacly as it happened with A3, this is sufficient to compute that the output is 0, and the ouput of N4 is 1. In this case, we still have the inconsistency with observation $out(\mathsf{N4}) = 0$, thus $\{\mathsf{O1}\}$ is *not* a diagnosis.

As the reader can easily verify, the complete set of diagnoses for this diagnostic problem is the following:

$$\{\{\mathsf{N4}\}, \{\mathsf{A4}\}, \{\mathsf{A2}, \mathsf{O3}\}, \{\mathsf{A2}, \mathsf{N3}, \}, \{\mathsf{A2}, \mathsf{A3}\}, \{\mathsf{A2}, \mathsf{O2}, \}, \{\mathsf{A2}, \mathsf{O1}\}\} \quad ★$$

2.2 Diagnoses and conflict sets

The characterization of consistency-based diagnosis that makes use of the notion of conflict set gives some insight on the notion of diagnosis itself, and is often found in the literature. Two very good reasons for discussing it here.
A conflict set is a set of components that, given the observations, cannot be simultaneously correct.

Definition 9 A set of components $\{c_1, \ldots, c_k\} \subseteq \mathsf{COMPS}$ is a **conflict set** for a diagnostic problem DP if $\mathsf{SD} \cup \mathsf{OBS} \cup \{ok(c_1), \ldots, ok(c_k)\}$ is inconsistent. A conflict set is **minimal** if there is no other conflict set properly contained in it. ★

What a conflict set tells us is: *"at least one of these components must be faulty."*

Example 10 Let us consider again the diagnostic problem introduced in example 3. It is easy to see (and figure 3.b shows it quite clearly) that $\{\mathsf{A2}, \mathsf{A4}, \mathsf{N4}\}$ is a conflict set for this diagnostic problem. Without knowing anything on how the other components work, assuming that these components are ok is sufficient to derive a contradiction with observations. In fact, if A2 is working, since one of its inputs is 0 its output towards A4 is also 0. Again, since one of the outputs of A4 is 0, its outputs towards N4 is 0 as well. Thus, the output of N4 evaluates to 1, while it has been observed to be 0. ★

We can reformulate proposition 6 in conflict set language:

Proposition 11 $\Delta \subseteq \mathsf{COMPS}$ *is a consistency-based diagnosis for a diagnostic problem* DP *if and only if it is a minimal set such that* $\mathsf{COMPS} \setminus \Delta$ *is* not *a conflict set for* DP. ★

We said that a conflict set meaning is *"at least one of these components must be faulty."* Then every consistency-based diagnosis should have at least one element in common with each conflict set, and we should be able to build consistency-based diagnoses by combining elements from different conflict sets. The following notion captures this process:

Definition 12 Given a collection $\mathcal{C}$ of sets, a **hitting-set** for $\mathcal{C}$ is a set $H \subseteq \bigcup_{S \in \mathcal{C}} S$ such that $H \cap S \neq \varnothing$ for each $S \in \mathcal{C}$. A hitting set is **minimal** if no proper subset of it is a hitting set for $\mathcal{C}$. ★

We can now prove the relation between diagnoses and conflict sets we have informally introduced:

Proposition 13 $\Delta \subseteq \mathsf{COMPS}$ *is a consistency-based diagnosis for a diagnostic problem* DP *if and only if it is a minimal hitting set for the collection of conflict sets for* DP.

Proof. Again, we divide the proof in two parts.

i. *If* Δ *is a consistency-based diagnosis for* DP*, then it is a minimal hitting set for the collection of conflict sets of* DP.

   By proposition 11, we have that $\mathsf{COMPS} \setminus \Delta$ is not a conflict set for DP. Given a conflict set Γ for DP, Γ cannot be a subset of $\mathsf{COMPS} \setminus \Delta$ because otherwise $\mathsf{COMPS} \setminus \Delta$ would be a conflict set as well. Thus Γ must have at least one element in common with Δ, which means that Δ is a hitting set for the collections of conflict sets of DP. Now we need to show that Δ is a *minimal* hitting set. Let $\{c_1, \ldots, c_k\}$ denote the components in Δ, and let us define $C_i = \mathsf{COMPS} \setminus \Delta \cup \{c_i\}$. Every C_i is a conflict set: in fact since $\Delta' = \Delta \setminus \{c_i\}$ is a subset of Δ, if $\mathsf{COMPS} \setminus \Delta' \equiv C_i$ was a conflict set, proposition 11 would be contradicted. Moreover, every C_i has exactly one element (that is, c_i) in common with Δ. Thus any subset of Δ does not hit at least one of the C_i sets, and therefore cannot be a hitting set. Hence, Δ is a minimal hitting set.

ii. *If Δ is a minimal hitting set for the collection of conflict sets of DP, then Δ is a consistency-based diagnosis for DP.*

Again, we exploit proposition 11. In order to prove that Δ is a diagnosis, first we prove that $\mathsf{COMPS} \setminus \Delta$ is not a conflict set. Then we prove that for all $\Delta' \subset \Delta$, $\mathsf{COMPS} \setminus \Delta'$ is a conflict set.

The first part is straightforward. It is clear that Δ does not hit $\mathsf{COMPS} \setminus \Delta$, and since by hypothesis Δ hits all conflict sets, we have that $\mathsf{COMPS} \setminus \Delta$ cannot be a conflict set.

Now let us consider $\Delta' \subset \Delta$ and let $\{c_1, \ldots, c_m\}$ denote those elements that are in Δ but not in Δ'. Since Δ is a minimal hitting set, for each c_i there must be a conflict set C_i such that $C_i \cap \Delta \equiv \{c_i\}$ (otherwise $\Delta \setminus \{c_i\}$ would be a hitting set and Δ would not be minimal). Obviously, the union of several conflict sets is a conflict set as well, thus $C = \bigcup_{i=1,\ldots,m} C_i$ is a conflict set, and $C \cap \Delta = \{c_1, \ldots, c_m\}$. A superset of a conflict set is also a conflict set, hence $C \cup \mathsf{COMPS} \setminus \Delta$ is a conflict set, and moreover $C \cup \mathsf{COMPS} \setminus \Delta \equiv (\mathsf{COMPS} \setminus \Delta) \cup \{c_1, \ldots, c_m\} \equiv \mathsf{COMPS} \setminus \Delta'$. ★

When a hitting set hits a conflict set, it hits also all of its supersets, which are conflict sets as well. Hence, H is a minimal hitting set for the collection of all conflict sets if and only if it is a minimal hitting set for the collection of minimal conflict sets. Thus we have the following corollary:

Proposition 14 *$\Delta \subseteq$ COMPS is a consistency-based diagnosis for a diagnostic problem DP if and only if it is a minimal hitting set for the collection of* minimal *conflict sets for DP.* ★

Example 15 Let us find the collection of minimal conflict sets for the diagnostic problem introduced in example 3. We already mentioned the conflict set $\{\mathsf{A2}, \mathsf{A4}, \mathsf{N4}\}$: it is easy to see that it is minimal, since removing anyone of its element removes also the conflict. For example, it is possible that both A4 and N4 are correct while observing a 0 output from N4: it suffices to assume that both the inputs to A4 are wrong. Conversely, it is definitely a conflict assuming that A4, N4 *and* one of the inputs of A4 are correct. This leads to the following additional minimal conflict: $\{\mathsf{O1}, \mathsf{O2}, \mathsf{A3}, \mathsf{N3}, \mathsf{O3}, \mathsf{A4}, \mathsf{N4}\}$. It is easy to see that the characterization of diagnosis based on conflicts leads exactly to the diagnoses mentioned in example 8. ★

3 Diagnosis with failure modes

3.1 Abductive diagnosis

Abductive diagnosis captures common reasoning in medical diagnosis, where explaining a symptom usually means finding a set of causes that imply the symptom itself. An hypothesis thus *explains* a symptom not by being consistent with it, but by logically implying it. For this to be possible, however, the system model must describe what happens in presence of a failure. If we consider the component-oriented model we introduced in the previous sections, this means that each component type model must include a description of what can happen in case of a failure, possibly distinguishing different failure types.

In the case of abductive diagnosis, we need to modify the notion of **behaviour of a component type**. For each type of component t, we define a set of behaviour mode predicates $modes(t) = \{ok, f_1, \ldots, f_h\}$ where ok denotes the correct behaviour and $f_1, \ldots, f_h$ denote the possible failure modes. Then, for each behaviour mode $m \in modes(t)$ we describe the expected correlation between its inputs and outputs, under the assumption that the component is in mode m. All formulas describing the behaviour follow the template:

$$type_t(x) \wedge m(x) \rightarrow \Phi(x)$$

where $\Phi(x)$ is a generic formula, $type_i$ is a predicate stating that component x belongs to type t, and the predicate $m(x)$ with $m \in modes(t)$ states that the component x is in mode m.

Example 16 Let us consider again the circuit example introduced in the previous section. We can assume that each gate (whether it is an *and*, an *or* or a *not* gate) has two possible failure modes: *stuck0*, where it always outputs a 0, and *stuck1*, where it always outputs a 1. Let us explicitly describe these behaviour modes for the *and* gate (the corresponding behaviour modes for the *or* and *not* gates are almost identical):

$$\begin{aligned}
(and_g(x) \wedge stuck0(x)) \rightarrow \quad & (in_1(x) = 0 \vee in_1(x) = 1) \wedge \\
& (in_2(x) = 0 \vee in_2(x) = 1) \wedge \\
& out(x) = 0 \\
(and_g(x) \wedge stuck1(x)) \rightarrow \quad & (in_1(x) = 0 \vee in_1(x) = 1) \wedge \\
& (in_2(x) = 0 \vee in_2(x) = 1) \wedge \\
& out(x) = 1 \qquad \bigstar
\end{aligned}$$

Within this context, diagnoses are not simply sets of components that are assumed to be incorrect, but they are rather assignments of behaviour modes to components. The following definition formalizes this notion:

Definition 17 Let SD be a system model with behaviour modes, and for each component c in the system let $modes(c)$ denote the behaviour mode predicates associated to the component type of c (that is, $modes(c) = modes(t)$ if $t(c)$ is a type declaration in SD). A **mode assignment** for SD is a set of predicates $M = \{m_1(c_1), \ldots, m_n(c_n)\}$ where $\{c_1, \ldots, c_n\}$ is the set of components in SD and $m_i \in modes(c_i)$ for each $i = 1, \ldots, n$. $\bigstar$

A mode assignment thus assigns exactly one mode to each component of the system. In the case of abductive diagnosis, then, the search space of the diagnostic algorithm is the set of all possible mode assignments:

$$\{m(c_1) \mid m \in modes(c_1)\} \times \ldots \times \{m(c_n) \mid m \in modes(c_n)\}$$

One could be tempted to use the same definition of diagnositc problem as we did for consiistency-based diagnosis, and to modify the definition of diagnosis as follows:

> Let $\mathsf{DP} = \langle \mathsf{SD}, \mathsf{COMPS}, \mathsf{OBS} \rangle$ be a diagnostic problem. We say that a mode assignment M is an **abductive diagnosis** for DP if $\mathsf{SD} \cup M \models \mathsf{OBS}$. $\bigstar$

Unfortunately, this does not work in general: in fact, OBS contains generic observations about the system, but not all of these can be charachterized as *symptoms*. An observation is a *symptom* if it is a *consequence* of the current behaviour mode of the system. Not all observations are consequences. If we consider our circuit example, we see that OBS contains both observations on system inputs and observations on system outputs. Certainly we cannot regard observations on inputs as symptoms, and we cannot expect the system model to *imply* them. Rather, they should be *added* to the system model in order for it to imply something on the outputs!
This leads to the following remarks:

1. A system description and a mode assignment alone could not be enough to *imply* the symptoms; some additional information about the *context* of system operation (e.g. the values of system inputs) may be needed.

2. Like observations, symptoms are ground atomic formulas. Unlike observations, however, symptoms should be restricted to those predicates that actually express *consequences* of system operation.

Therefore we have to redefine the notion of diagnostic problem as follows:

Definition 18 Let SD be a system description with behaviour modes, where we distinguish two types of predicates: **context** predicates, describing the operational context of the system, and **symptom** predicates, describing observable consequences of system operation. Then an **abductive diagnostic problem** is a quadruple $\langle \mathsf{SD}, \mathsf{COMPS}, \mathsf{CTX}, \mathsf{SMP} \rangle$ such that:

- SD is the system description;
- COMPS is the set of system components;
- CXT is a set of atomic ground formulas over context predicates;
- SMP is a set of atomic ground formulas over symptom predicates;
- $\mathsf{SD} \cup \{ok(c) \mid c \in \mathsf{COMPS}\} \cup \mathsf{CXT} \cup \mathsf{SMP}$ is inconsistent. ★

We can now give the definition of abductive diagnosis:

Definition 19 Let DP be an abductive diagnostic problem with system description SD. A mode assignment M over SD is an **abductive diagnosis** for DP if $\mathsf{SD} \cup M \cup \mathsf{CXT}$ is consistent and $\mathsf{SD} \cup M \cup \mathsf{CXT} \models \mathsf{SMP}$. ★

The requirement for $\mathsf{SD} \cup M \cup \mathsf{CXT}$ to be consistent is a theoretical one, in the sense that, in practice, context predicates should not be constrained by the model, and therefore any CXT should be consistent with the system for any given mode assignment.

It is possible (although not strictly necessary) to define an ordering relation (and thus a notion of minimality) also for abductive diagnoses. In general, minimal diagnoses are always preferred over non-minimal ones (notice however that here we have a notion of non-minimal diagnoses, while in the consistency based approach all diagnoses are minimal).

Definition 20 Let M and M' be two different mode assignments for the same system SD. We say that M is **simpler** than M', and write $M < M'$, if $\{m(c) \in M \mid m \neq ok\} \subset \{m(c) \in M' \mid m \neq ok\}$. ★

Example 21 Let us consider the diagnostic problem we introduced in example 3. We can transform it into an abductive diagnostic problem by deciding that CXT contains observations on inputs and SMP observations on outputs:

$$\begin{array}{rlll} \mathsf{CXT} = & \{ & in_1(\mathsf{A1}) = 0, & in(\mathsf{N1}) = 1, \\ & & in_2(\mathsf{O1}) = 1, & in_2(\mathsf{A2}) = 0 \quad \} \\ \mathsf{SMP} = & \{ & out(\mathsf{O4}) = 1, & out(\mathsf{N4}) = 0 \quad \} \end{array}$$

Now, let us find some (possibly minimal) abductive diagnoses for our diagnostic problem. We can for example start from the consistency-based diagnoses we computed in example 8, and see whether it is possible to turn them into abductive diagnoses by assigning a specific failure mode to each component.
For example, starting from the consistency-based diagnosis {A4} we can argue that in order to observe a 0 in the output of N4 we need to have a 1 as output of A4. Thus {*stuck1*(A4)} is an abductive diagnosis for our diagnostic problem.[1]
Now let us see what happens if we assume that A2 and O1 are faulty. In order to observe a 0 in the output of N4 we need both inputs of A4 to be 1. So, first of all, we need to assume that A2 is in *stuck1* mode. Then, we need one of the inputs to O3 to be 1. The first way may be to obtain an 1 as output of A3, but this is not possible, because one of the inputs of A3 is set to 0 by the context. Then, we need to have a 1 as output of N3, which means having a 0 as output of O2. If we assume that O1 is in *stuck0* mode we obtain

[1] In describing diagnoses we do not list explicitly the *ok* assignments, since they can be derived from the others.

precisely this effect. Unfortunately, however, we also obtain that the output of O4 is 0, while it has been observed to be 1. The only way to have back a 1 as output of O4 is to assume that A1 is in *stuck1* mode. Therefore, another abductive diagnosis for our diagnostic problem is $\{stuck1(A2), stuck0(O1), stuck1(A1)\}$. The complete set of abductive diagnoses for out diagnostic problem is the following:

$$
\begin{array}{ll}
\{\ \{stuck0(N4)\}, & \{stuck1(A4)\}, \\
\ \{stuck1(A2), stuck1(O3)\}, & \{stuck1(A2), stuck1(N3)\}, \\
\ \{stuck1(A2), stuck1(A3)\}, & \{stuck1(A2), stuck0(O2)\}, \\
\ \{stuck1(A2), stuck1(O1), stuck1(O4)\}, & \{stuck1(A2), stuck1(O1), stuck1(A1)\}\ \ \}
\end{array}
$$

Exercise. *As we have seen, it is possible that a consistency-based diagnosis as defined in section 2.1 cannot be turned into an abductive diagnosis, and viceversa. Then does it make sense to start from consistency-based diagnoses to find abductive ones? And why is or isn't it so?* ★

3.2 Abductive diagnosis vs. consistency-based diagnosis

Pure consistency-based diagnosis assumes models that describe only the correct behaviour of a system. However, it is possible to define consistency-based diagnosis also on models with behaviour modes.

Consistency-based diagnosis does not distinguish context and symptom predicates; therefore it is straightforward to convert an abductive diagnostic problem into a standard diagnostic problem, by simply defining $\mathsf{OBS} = \mathsf{CXT} \cup \mathsf{SMP}$ and discarding from the model information regarding the two predicate types. Hence, it is possible to define the notion of consistency-based diagnosis for an abductive diagnostic problem.[2]

Definition 22 Let DP be an abductive diagnostic problem with system description SD. A mode assignment M over SD is a **consistecy-based diagnosis** for DP if $\mathsf{SD} \cup M \cup \mathsf{CXT} \cup \mathsf{SMP}$ is consistent. ★

We then have the following property:

Proposition 23 *Given an abductive diagnostic problem DP, if M is an abductive diagnosis for DP then M is also a consistency-based diagnosis for DP.*

Proof. Let us assume M is an abductive diagnosis but not a consistency base diagnosis. Then we have:

(1) $\mathsf{SD} \cup M \cup \mathsf{CXT}$ is consistent

(2) $\mathsf{SD} \cup M \cup \mathsf{CXT} \models \mathsf{SMP}$

(3) $\mathsf{SD} \cup M \cup \mathsf{CXT} \cup \mathsf{SMP}$ is inconsistent

But if $\mathsf{SD} \cup M \cup \mathsf{CXT}$ is inconsistent with its own implications ((2),(3)) then $\mathsf{SD} \cup M \cup \mathsf{CXT}$ itself is inconsistent, thereby contradicting (1). ★

Therefore, in general, the set of abductive diagnoses is a superset of the set of consistency-based diagnoses with failure modes. Are there cases where the two coincide? The answer is yes, and those cases can be characterized as follows.

Definition 24 We say that a system description SD is **complete** with respect to a context CXT iand a set of symptoms SMP f for each mode assignment M and for each ground atomic formula $\sigma \in \mathsf{SMP}$either $\mathsf{SD} \cup M \cup \mathsf{CXT} \models \sigma$ or $\mathsf{SD} \cup M \cup \mathsf{CXT} \models \neg\sigma$. ★

Proposition 25 *Let $DP = \langle SD, COMPS, CXT, SMP\rangle$ be an abductive diagnostic problem. If SD is complete with respect to CXT and SMP then each consistency-based diagnosis with failure modes for DP is also an abductive diagnosis.*

[2] In order to compare the two definitions, it is useful to base them over the same notion of diagnostic problem.

Proof. Let σ be a symptom, and let M be a consistency-based diagnosis. By definition of consistency-based diagnosis with failure modes we have that $\mathsf{SD} \cup M \cup \mathsf{CXT} \cup \mathsf{SMP}$ is consistent. Thus it is not possible that $\mathsf{SD} \cup M \cup \mathsf{CXT} \models \neg\sigma$, and, by definition of completeness, it must be that $\mathsf{SD} \cup M \cup \mathsf{CXT} \models \sigma$. Thus $\mathsf{SD} \cup M \cup \mathsf{CXT} \models \mathsf{SMP}$. ★

Example 26 Let us consider once more the circuit example. The circuit model is such that, once all of its inputs are fixed, all the values of the other inputs and outputs can be uniquely determined. So for every contect CXT that assigns a value to all the system inputs, and for every possible set of symptoms SMP, the system description is complete with respect to CXT and SMP. Thus, if we take the abductive diagnostic problem defined in example 21, we can easily see that the set of abductuve diagnoses coincides with the set of consistency-based diagnoses with failure modes.
In order to find a diagnostic problem for the circuit example where the two sets do not coincide, we need to specify a context that does not assign a value to every input.
Exercise. *Modify the diagnostic problem in example 21 so that the context is* $\mathsf{CXT} = \{in_1(\mathsf{A1}) = 0, in_2(\mathsf{A2}) = 0\}$. *Show that the result is still a diagnostic problem, and show that for the new diagnostic problem there are consistency-based diagnoses with failure modes that are not abductive diagnoses.* ★

There are several weakenings of the basic notion of abductive diagnosis that bring it closer to the consisency-based approach. We will not discuss them here; it suffices to mention that depending on the features of a diagnostic system and/or class of models, a slightly different notion of diagnosis may be adopted to obtain the desired results.
Another element that influences the diagnostic process is how to choose *preferred* diagnoses. As we have seen, a diagnostic system may return several candidate diagnoses, and there are different criteria for deciding which one is most likely to be **the** diagnosis. We already mentioned the cardinality criterion: choosing diagnoses with the lowest number of failed components. Other criteria might be related to probabilities or other considerations depending on the specific model class.

4 Diagnosis of physical systems

4.1 Qualitative modelling

One of the main applications of model-based diagnosis are physical systems (e.g. the engine of a car, the electrical equipment of an airplane, the pneumatic machinery of an industrial plant...). However, physical systems are *continuous* while the notions we have introduced so far apply essentially to discrete systems.
In the model-based diagnosis literature there are two main approaches for giving a discrete model of a physical system:

- **Discrete-event models.** In this case the system is seen as an evolving process, and we do not model physical quantities, but events that happen on the system (e.g. *the pipe breaks* or *the valve closes* or *the temperature reaches a fixed threshold*). The model of the system is in this case a finite-state automata. We will not discuss in detail this approach.
- **Qualitative reasoning.** This approach is born in the field of AI, and the idea behind it is that human reasoning on physical systems (and especially diagnostic reasoning) usuallly does not need precise measures over the system, but rather works in terms of *too high/too low* or *present/absent* or, more generally, in terms of broad ranges of values. Thus the system is described in terms of physical quantities, but variables range over a finite set of values, each representing a range of real values. In the remaining part of this section we will follow this approach.

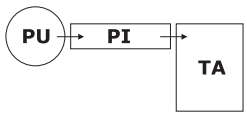

Figure 4: A simple physical system.

Basically, qualitative models for diagnosis follow the same pattern as we saw in the previous section: we have component types, component instances and a system structure. However, each component type model is the qualitative description of its physical counterpart. Let us see an example.

Example 27 The system we want to model is depicted in figure 4. Its components are a pump, a pipe and a tank. The goal of the system is to fill the tank by pumping liquid in it (obviously this is only a partial view of the system, since the pump will have to take the liquid somewhere else). One of the main difficulties in writing a qualitative model is choosing the proper level of abstraction. Since this is an example, we will abstract a lot.
Let us start by modelling the pump: it is charachterized by two quantities, *command* and *outflow*. The former represents the command to the pump (*on* or *off*) while the latter represents the flow coming out from the pump (*normal*, *low* or *absent*). We assume that the pump can have a *broken* failure mode, where it does not produce any flow even if turned on. Here is the model:

$$\begin{array}{lcl} pump(x) \wedge ok(x) & \rightarrow & (command(x) = on \vee command(x) = off) \wedge \\ & & (command(x) = on \rightarrow outflow(x) = normal) \wedge \\ & & (command(x) = off \rightarrow outflow(x) = absent) \\ pump(x) \wedge broken(x) & \rightarrow & (command(x) = on \vee command(x) = off) \wedge \\ & & (outflow(x) = absent) \end{array}$$

Now let us consider the pipe. Here the interesting quantities are the input flow (*inflow*) and the output flow (*outflow*), which are usually the same. However, the pipe may be *leaking*: in this case the output flow may be less than it should be.

$$\begin{array}{lcl} pipe(x) \wedge ok(x) & \rightarrow & (inflow(x) = normal \vee inflow(x) = low \vee inflow(x) = absent) \wedge \\ & & (outflow(x) = inflow(x)) \\ pipe(x) \wedge leaking(x) & \rightarrow & (inflow(x) = normal \vee inflow(x) = low \vee inflow(x) = absent) \wedge \\ & & ((inflow(x) = normal \vee inflow(x) = low) \rightarrow outflow(x) = low) \wedge \\ & & (inflow(x) = absent \rightarrow outflow(x) = absent) \end{array}$$

Finally, let us consider the tank. The tank has two relevant quantities: the input flow (*inflow*), that can be *normal*, *low* or *absent* and the level of the liquid after receiving the input flow (*level*), which can be *empty*, *intermediate* or *full*. We assume that initially the level is *intermediate*, and that a *normal* flow would fill up the tank, while a *low* flow may fill it or not. Moreover, the tank can be *leaking*. In this case with a *normal* flow the tank may fill or not, but the level does not decrease. With a *low* flow the level may also decrease, and with an *absent* flow the level decreases for sure.

$$\begin{array}{lcl} tank(x) \wedge ok(x) & \rightarrow & (inflow(x) = normal \vee inflow(x) = low \vee inflow(x) = absent) \wedge \\ & & (inflow(x) = normal \rightarrow level(x) = full) \wedge \\ & & (inflow(x) = low \rightarrow (level(x) = full \vee level(x) = intermediate)) \wedge \\ & & (inflow(x) = absent \rightarrow level(x) = intermediate) \\ tank(x) \wedge leaking(x) & \rightarrow & (inflow(x) = normal \vee inflow(x) = low \vee inflow(x) = absent) \wedge \\ & & (inflow(x) = normal \rightarrow (level(x) = full \vee level(x) = intermediate)) \wedge \\ & & (inflow(x) = low \rightarrow (level(x) = full \vee level(x) = intermediate \vee level(x) = empty)) \wedge \\ & & (inflow(x) = absent \rightarrow (level(x) = intermediate \vee level(x) = empty)) \end{array}$$

In order to complete the model, we need to add component instantiations:$\{pump(\mathsf{PU}), pipe(\mathsf{PI}), tank(\mathsf{TA})\}$, and connections: $\{outflow(\mathsf{PU}) = inflow(\mathsf{PI}), outflow(\mathsf{PI}) = inflow(\mathsf{TA})\}$. Our model is now finished and ready for diagnosis. ★

Frequently, qualitative models are non-deterministic, in the sense that it is not always possible to determine the value of an output variable given the inputs and a mode assignment. If you look at the simple example above, you can easily see that when the pump is on and both the pipe and the tank are leaking it is not possible to tell what the level in the tank will be. Of course non-determinism is due to abstraction: in order to know what happens, we would need to know exactly the amount of liquid that goes lost due to the leakeage. From the point of view of diagnosis this is not always a problem. In fact, if we observe that the pump is on but the tank is empty we are able to conclude that the tank must be leaking. Notice however that the kind of diagnostic reasoning we performed follows the consistency-based approach rather than the abductive one: in fact, due to non-determinism, abductive diagnosis is too restrictive for this kind of models.

4.2 Diagnosis as constraint satisfaction

Qualitative models are finite: this means that it is possible to list all possible "worlds" in which the formulas describing a system are true, and these possible worlds can be given as sets of ground atomic formulas that assign a unique behaviour mode to each component and a unique value to each system quantity. Then, given a diagnostic problem $\langle \mathsf{SD}, \mathsf{COMPS}, \mathsf{OBS} \rangle$ we can *simulate* the system by building all possible worlds $\{W_1, \ldots, W_k\}$ where $\mathsf{SD} \cup \mathsf{OBS}$ holds, and deriving from each world W_i a diagnosis M_i as follows:

$$M_i = \{m(c) \in W_i \mid m \text{ is a behaviour mode predicate}, c \in \mathsf{COMPS}\}$$

It is straightforward to see that this corresponds to consistency-based diagnosis with failure modes.

Example 28 Let us consider the diagnostic problem $\langle \mathsf{SD}, \mathsf{COMPS}, \mathsf{OBS} \rangle$ where SD and COMPS correspond to the system described in the previous example, and $\mathsf{OBS} = \{\mathit{command}(\mathsf{PU}) = \mathit{on}, \mathit{level}(\mathsf{TA}) = \mathit{empty}\}$.
The possible worlds are the following:

$$\begin{array}{lcll}
W_1 & = & \{ & \mathit{command}(\mathsf{PU}) = \mathit{on}, \quad \mathit{ok}(\mathsf{PU}), \quad \mathit{outflow}(\mathsf{PU}) = \mathit{normal}, \\
 & & & \mathit{inflow}(\mathsf{PI}) = \mathit{normal}, \quad \mathit{leaking}(\mathsf{PI}), \quad \mathit{outflow}(\mathsf{PI}) = \mathit{low}, \\
 & & & \mathit{inflow}(\mathsf{TA}) = \mathit{low}, \quad \mathit{leaking}(\mathsf{TA}), \quad \mathit{level}(\mathsf{TA}) = \mathit{empty}, \quad \} \\
W_2 & = & \{ & \mathit{command}(\mathsf{PU}) = \mathit{on}, \quad \mathit{broken}(\mathsf{PU}), \quad \mathit{outflow}(\mathsf{PU}) = \mathit{absent}, \\
 & & & \mathit{inflow}(\mathsf{PI}) = \mathit{absent}, \quad \mathit{ok}(\mathsf{PI}), \quad \mathit{outflow}(\mathsf{PI}) = \mathit{absent}, \\
 & & & \mathit{inflow}(\mathsf{TA}) = \mathit{absent}, \quad \mathit{leaking}(\mathsf{TA}), \quad \mathit{level}(\mathsf{TA}) = \mathit{empty}, \quad \} \\
W_3 & = & \{ & \mathit{command}(\mathsf{PU}) = \mathit{on}, \quad \mathit{broken}(\mathsf{PU}), \quad \mathit{outflow}(\mathsf{PU}) = \mathit{absent}, \\
 & & & \mathit{inflow}(\mathsf{PI}) = \mathit{absent}, \quad \mathit{leaking}(\mathsf{PI}), \quad \mathit{outflow}(\mathsf{PI}) = \mathit{absent}, \\
 & & & \mathit{inflow}(\mathsf{TA}) = \mathit{absent}, \quad \mathit{leaking}(\mathsf{TA}), \quad \mathit{level}(\mathsf{TA}) = \mathit{empty}, \quad \}
\end{array}$$

We thus obtain the following diagnoses:

$$\begin{array}{lcl}
M_1 & = & \{\mathit{leaking}(\mathsf{PI}), \mathit{leaking}(\mathsf{TA})\} \\
M_2 & = & \{\mathit{broken}(\mathsf{PU}), \mathit{leaking}(\mathsf{TA})\} \\
M_3 & = & \{\mathit{broken}(\mathsf{PU}), \mathit{leaking}(\mathsf{PI}), \mathit{leaking}(\mathsf{TA})\}
\end{array}$$

M_3 can be discarded because it is not minimal. ★

What we apply in the case of qualitative models is the well known *closed-world assumption*. This implies that the model can be equivalently expressed in propositional logic (rather than in first-order logic), and also that the model can be described as a *constraint satisfaction problem*.

Definition 29 Let $\mathbb{X} = X_1, \ldots, X_n$ be a set of variables over finite domains $D_1, \ldots, D_n$. A **constraint** $C(X_{i_1}, \ldots, X_{i_k})$ is a relation over $D_{i_1} \times \ldots \times D_{i_k}$. Let α denote an assignment of values to $X_1, \ldots, X_n$ such that $\alpha(X_i) \in D_i$. We say that α satisfies C if $(\alpha(X_{i_1}), \ldots, \alpha(X_{i_k})) \in C$.

Given a set of constraints $\mathbb{C} = C_1, \ldots, C_m$ over a set of variables $\mathbb{X}$, the **constraint satisfaction problem** consists in finding an assignment α for $\mathbb{X}$ that satisfies all constraints in $\mathbb{C}$. A variant of this problem consists in finding *all* assignments α that satisfy $\mathbb{C}$.

A qualitative model can be turned into a set of constraints. Each **component type** can be represented as a constraint, whose variables are the relevant quantities involved in the component description, plus a variable representing the behaviour mode. The model consists in the set of all combinations of values for such variables that are correct with respect to the component physical behaviour. **Component instances** are created by introducing a set of unique variables for each instance, and by applying the proper type constraint to those variables. **System structure** is specified by adding equality constraints between variables of different component types.
If we consider a diagnostic problem $\mathsf{DP} = \langle \mathsf{SD}, \mathsf{COMPS}, \mathsf{OBS} \rangle$, also observations can be represented as equality constraints. The constraint problem corresponding to DP is given by $\mathsf{SD} \cup \mathsf{OBS}$. Given a solution α to this constraint problem we can turn it into a consistency-based diagnosis by restricting it to behaviour mode variables.

Example 30 Let us try to turn the model of our simple hydraulic system into a constraint problem. For the **pump** component type we can introduce a constraint over three variables: x, representing the command, y, representing the output flow, and m, representing the behaviour mode:

Constraint **pump**		
m	x	y
ok	*on*	*normal*
ok	*off*	*absent*
broken	*on*	*absent*
broken	*off*	*absent*

In a similar way, we describe the **pipe** component type over three variables, where m denotes again the behaviour mode, x the input flow, and y the output flow.

Constraint **pipe**		
m	x	y
ok	*normal*	*normal*
ok	*low*	*low*
ok	*absent*	*absent*
leaking	*normal*	*low*
leaking	*low*	*low*
leaking	*absent*	*absent*

Finally, we describe the **tank** component type, where m denotes as usual the behaviour mode, x the input flow, and y the level of liquid.

Constraint **tank**		
m	x	y
ok	*normal*	*full*
ok	*low*	*full*
ok	*low*	*intermediate*
ok	*absent*	*intermediate*
leaking	*normal*	*full*
leaking	*normal*	*intermediate*
leaking	*low*	*full*
leaking	*low*	*intermediate*
leaking	*low*	*empty*
leaking	*absent*	*intermediate*
leaking	*absent*	*empty*

In order to describe the system, we first of all need to instantiate the components. We create a specific triple of variables for each instance: $\langle m_{\mathsf{PU}}, x_{\mathsf{PU}}, y_{\mathsf{PU}}\rangle$ for the pump, $\langle m_{\mathsf{PI}}, x_{\mathsf{PI}}, y_{\mathsf{PI}}\rangle$ for the pipe, and $\langle m_{\mathsf{TA}}, x_{\mathsf{TA}}, y_{\mathsf{TA}}\rangle$ for the tank. Then we define the set of constraints:

$$\mathbb{C}_0 = \{\mathbf{pump}(m_{\mathsf{PU}}, x_{\mathsf{PU}}, y_{\mathsf{PU}}), \mathbf{pipe}(m_{\mathsf{PI}}, x_{\mathsf{PI}}, y_{\mathsf{PI}}), \mathbf{tank}(\langle m_{\mathsf{TA}}, x_{\mathsf{TA}}, y_{\mathsf{TA}})\}$$

Then we need to specify connections; this can easily be done with the following set of constraints: $\mathbb{C}_1 = \{y_{\mathsf{PU}} = x_{\mathsf{PI}}, y_{\mathsf{PI}} = x_{\mathsf{TA}}\}$. Thus the system description is $\mathsf{SD} = \mathbb{C}_0 \cup \mathbb{C}_1$.
If we consider again the diagnostic problem introduced in the previous example, we can describe observations as constraints: $\mathsf{OBS} = \{x_{\mathsf{PU}} = \mathit{on}, y_{\mathsf{TA}} = \mathit{empty}\}$. It is easy to see that the constraint problem $\mathsf{SD} \cup \mathsf{OBS}$ has three solutions, that correspond exactly to the three possible worlds we computed in the previous example. ★

It is worth noting that constraint satisfaction (and thus consistency-based diagnosis of finite systems) is, in general, an NP-hard problem, although in some cases it is possible to make assumptions that make the problem easier to solve.

References

[1] T. Bylander. Some causal models are deeper than others. *Artificial Intelligence in Medicine*, 2(3):123–128, 1990.

[2] B. Chandrasekaran, J.W. Smith, and J. Sticklen. Deep models and their relation to diagnosis. *Artificial Intelligence in Medicine*, 1(1):29–40, 1989.

[3] W.J. Clancey and R. Letsinger. NEOMYCIN: Reconfiguring a rule-based expert system for application to teaching. In *Proc 7th IJCAI*, pages 829–836, Vancouver, 1981.

[4] L. Console, D. Theseider Dupré, and P. Torasso. A theory of diagnosis for incomplete causal models. In *Proc. 11th IJCAI*, pages 1311–1317, Detroit, 1989.

[5] L. Console, D. Theseider Dupré, and P. Torasso. On the relationship between abduction and deduction. *Journal of Logic and Computation*, 1(5):661–690, 1991.

[6] L. Console and P. Torasso. Hypothetical reasoning in causal models. *International Journal of Intelligent Systems*, 5(1):83–124, 1990.

[7] L. Console and P. Torasso. A spectrum of logical definitions of model-based diagnosis. *Computational Intelligence*, 7(3):133–141, 1991.

[8] P.T. Cox and T. Pietrzykowski. General diagnosis by abductive inference. In *Proc. IEEE Symposium on Logic Programming*, pages 183–189, San Francisco, 1987.

[9] R. Davis. Expert systems: Where are we? And where do we go from here? *AI Magazine*, 3(2):3–22, 1982.

[10] R. Davis. Diagnostic reasoning based on structure and behavior. *Artificial Intelligence*, 24(1-3):347–410, 1984.

[11] J. de Kleer. Local methods for localizing faults in electronic circuits. Technical Report 394, MIT, AI, laboratory for Computer Science, 1976. Out of print.

[12] J. de Kleer, A. Mackworth, and R. Reiter. Characterizing diagnoses and systems. *Artificial Intelligence*, 56(2–3):197–222, 1992.

[13] R. Milne (ed.). Special issue on causal and diagnostic reasoning. *IEEE Trans. on Systems, Man and Cybernetics*, 17(3), 1987.

[14] B. El Ayeb, P. Marquis, and M. Rusinowitch. Deductive/abductive diagnosis: the da principle. In *Proceedings of the 9th European Conference on Artificial Intelligence (ECAI1990)*, pages 47–52, 1990.

[15] M.R. Genesereth. The use of design descriptions in automated diagnosis. *Artificial Intelligence*, 24(1-3):411–436, 1984.

[16] W. Hamscher, L. Console, and J. de Kleer, editors. *Readings in Model-Based Diagnosis*. Morgan Kaufmann, 1992.

[17] B. Kuipers and J. Kassirer. Causal reasoning in medicine: Analysis of a protocol. *Cognitive Science*, 8:363–385, 1984.

[18] D. Partridge. The scope and limitation of first generation expert systems. *Future Generation Computer Systems*, 3(1):1–10, 1987.

[19] R. Patil. Causal representation of patient illness for electrolyte and acid-base diagnosis. Technical Report LCS-267, MIT, Cambridge, MA, 1981.

[20] Y. Peng and J. Reggia. A probabilistic causal model for diagnostic problem solving - part II: Diagnostic strategy. *IEEE Trans. on Systems, Man and Cybernetics*, 17(3):395–406, 1987.

[21] Y. Peng and J. Reggia. *Abductive inference models for diagnostic problem solving*. Springer-Verlag, 1991.

[22] D. Poole. Representing knowledge for logic-based diagnosis. In *Proc. of the Int. Conf. on Fifth Generation Computer Systems*, pages 1282–1290, Tokyo, 1988.

[23] J.A. Reggia, D.S. Nau, and P.Y. Wang. Diagnostic expert systems based on a set covering model. *Int. J. of Man-Machine Studies*, 19(5):437–460, 1983.

[24] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–96, 1987.

[25] M. Shirley and R. Davis. Generating distinguishing tests based on hierarchical models and symptom information. In *Proc Int'l Conference on Computer Design*, 1983.

[26] P. Struss and O. Dressler. Physical negation - integrating fault models into the general diagnostic engine. In *Proc. 11th IJCAI*, pages 1318–1323, Detroit, 1989.

[27] P. Torasso and L. Console. Causal reasoning in diagnostic expert systems. In *Proc. V Int. Conf. on Applications of Artificial Intelligence*, pages 598–605, Orlando, 1987.